
Human Activity Learning using Object Affordances from RGB-D Videos

Hema Swetha Koppula, Rudhir Gupta and Ashutosh Saxena

Department of Computer Science

Cornell University

Ithaca, NY 14850

{hema, rg495, asaxena}@cs.cornell.edu

Abstract

Human activities comprise several sub-activities performed in a sequence and involve interactions with various objects. This makes reasoning about the object affordances a central task for activity recognition. In this work, we consider the problem of jointly labeling the object affordances and human activities from RGB-D videos. We frame the problem as a Markov Random Field where the nodes represent objects and sub-activities, and the edges represent the relationships between object affordances, their relations with sub-activities, and their evolution over time. We formulate the learning problem using a structural SVM approach, where labeling over various alternate temporal segmentations are considered as latent variables. We tested our method on a dataset comprising 120 activity videos collected from four subjects, and obtained an end-to-end precision of 81.8% and recall of 80.0% for labeling the activities.

1 Introduction

In this paper, we present a learning algorithm that takes as input an RGB-D video (obtained from an inexpensive sensor such as Microsoft Kinect), and identifies the human activities taking place over long time periods (e.g., see Fig. 2). Most prior work in human activity detection has focussed on activity detection from still images or from 2D videos. Estimating the human pose is the primary focus of these works, and they consider activities taking place over shorter time scales (see Section 2). Having access to a 3D camera, which provides RGB-D videos, enables us to robustly estimate human poses and use this information for learning complex human activities.

Our focus in this work is to recognize complex human activities that take place over long time scales and that consist of a long sequence of sub-activities, such as *making cereal* and *arranging objects* in a room. For example, *making cereal* activity consists of around 12 sub-activities on average, which includes *reaching* the pitcher, *moving* the pitcher to the bowl, and then *pouring* the milk into the bowl. This proves to be a very challenging task given the variability across individuals in performing each sub-activity, and other environment induced conditions such as cluttered background and viewpoint changes. (See Fig. 1 for some examples.)

In most previous works, object detection and activity recognition have been addressed as separate tasks. Only recently, some works have shown that modeling mutual context is beneficial [1, 2]. The key idea in our work is to note that, in activity detection, it is sometimes more informative to know *how* an object is being used (associated affordances, [3]) rather than knowing *what* the object is (i.e., the object category). For example, both chair and sofa might be categorized as ‘sittable,’ and a cup might be categorized as both ‘drinkable’ and ‘pourable.’ Note that the affordances of an object change over time depending on its use, e.g., a pitcher may first be *reachable*, then *movable* and finally *pourable*. In addition to helping activity recognition, recognizing object affordances is important by itself because of their use in robotic applications (e.g., [4]).

We propose a method to learn human activities by modeling the sub-activities and affordances of the objects, how they change over time, and how they relate to each other. More formally, we define a Markov Random Field over two kinds of nodes: object and sub-activity nodes. The edges in the graph model the pairwise relations among interacting nodes, namely the object-object interactions,

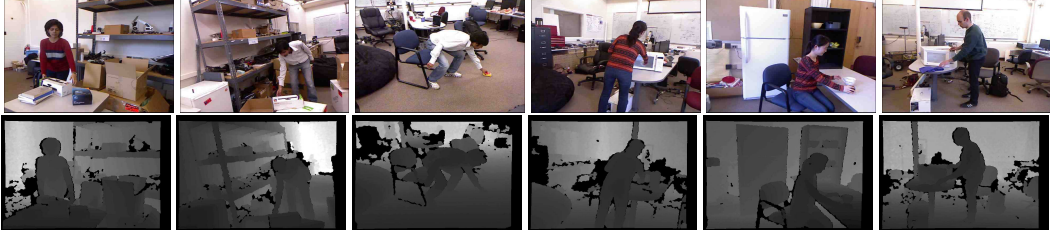


Figure 1: Example shots of *reaching* sub-activity from our dataset. Top row shows the RGB image, and the bottom row shows the corresponding depth images from the RGB-D camera. Note that there are significant variations in the way the subjects perform the sub-activity. In addition, there is significant background clutter and subjects are partially occluded (e.g., column 1) or not facing the camera (e.g., column 4) in many instances. object-sub-activity interaction, and the temporal interactions. This model is built with each spatio-temporal segment being a node. The parameters of this model are learnt using a structural SVM formulation [5]. Given a new sequence of frames, we label the high-level activity, all the sub-activities and the object affordances using our learned model.

However, the activities take place over a long time-scale, and different people execute sub-activities differently and for different periods of time. Furthermore, people also often merge two consecutive sub-activities together. Thus, segmentations in time are noisy and in fact, there may not be one ‘correct’ segmentation, especially at the boundaries. One approach could be to consider *all* possible segmentations, and marginalize the segmentation; however, this is computationally infeasible. In this work, we perform sampling of several segmentations, and consider labelings over these temporal segments as latent variables in our learning algorithm.

In extensive experiments over 120 activity videos collected from four subjects, we showed that our approach outperforms the baselines in both the tasks of activity as well as affordance detection. We achieved an accuracy of 91.8% for affordance, 86.0% for sub-activity labeling and 84.7% for high-level activities respectively when given the ground truth segmentation, and an end-to-end accuracy of 83.9%, 68.2% and 80.6% on these respective tasks.

2 Related Work

There has been a lot of work on human activity detection from images [6, 7] and from videos [8, 9, 10, 11, 12, 13, 14, 15, 16]. Here, we discuss works that are closely related to ours, and refer the reader to [17] for a survey of the field. Most works (e.g., [11, 12, 13]) consider detecting actions at a ‘sub-activity’ level (e.g., *walk*, *bend*, and *draw*) instead of considering high-level activities. Their methods range from discriminative learning techniques for joint segmentation and recognition [11, 12] to combining multiple models [13]. Some works such as [14] consider high-level activities. Tang et. al. [14] propose a latent model for high-level activity classification and have the advantage of requiring only high-level activity labels for learning. None of these methods explicitly consider the role of objects or object affordances that not only help in identifying sub-activities and high-level activities, but are also important for several robotic applications (e.g., [4]).

Some recent works [1, 2, 18, 19, 16] show that modeling the interaction between human poses and objects in 2D videos results in a better performance on the tasks of object detection and activity recognition. However, these works cannot capture the rich 3D relations between the activities and objects, and are also fundamentally limited by the quality of the human pose inferred from the 2D data. More importantly, for activity recognition, the object *affordance* matters more than its category.

Kjellstrom et. al. [20] used a Factorial CRF to simultaneously segment and classify human hand actions, as well as classify the object affordances involved in the activity from 2D videos. However, this work is limited to classifying only hand actions and does not model interactions between the objects. We consider complex full-body activities and show that modeling object-object interactions is important as objects have affordances even if they are not directly interacted with human hands.

Recently, with the availability of inexpensive RGB-D sensors, some works consider labeling and object recognition in 3D point-clouds [21, 22]. Sung et. al. [23] considered activity recognition from RGB-D videos. They propose a hierarchical maximum entropy Markov model to detect activities from RGB-D videos and treat the sub-activities as hidden nodes in their model. However, they use only human pose information for detecting activities and also constrain the number of sub-activities in each activity. In contrast, we model context from object interactions along with human pose, also present a better learning algorithm. (See Section 4 for further comparisons.) Gall et. al. [24] also use depth data to perform sub-activity (referred to as action) classification and functional categorization of objects. Their method first detects the sub-activity being performed using the estimated human pose from depth data, and then performs object localization and clustering of the objects into func-

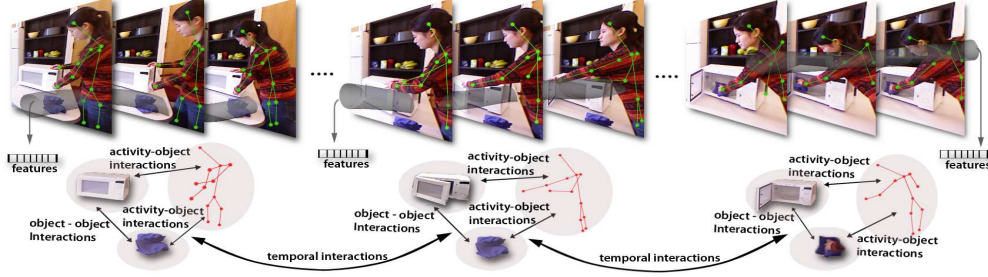


Figure 2: Pictorial representation of the different types of nodes and relationships modeled in part of the *cleaning objects* activity comprising three sub-activities: *reaching*, *opening* and *scrubbing*.

tional categories based on the detected sub-activity. In contrast, our proposed method performs joint sub-activity and affordance labeling and uses these labels to perform high-level activity detection.

All above works lack a unified framework of combining all the information available in human interaction activities and therefore we propose a model that captures both the spatial and temporal relations between objects and human poses to perform joint object affordance and activity detection.

3 Our Approach

Our goal is to perform joint activity and object affordance labeling of RGBD videos. As illustrated in Fig. 2, we define a Markov Random Field (MRF) over the spatio-temporal sequence we get from an RGBD video. The MRF is represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. There are two types of nodes in \mathcal{G} : objects nodes denoted by \mathcal{V}_o and sub-activity nodes denoted by \mathcal{V}_a .

3.1 Representation

If we build our graph with nodes for objects and sub-activities for each time instant (at 30fps), then we will end up with quite a large graph. Furthermore, such a graph would not be able to model meaningful transitions between the sub-activities because they take place over a long-time (e.g., a few seconds). Therefore, in our approach we first segment the video into small temporal segments, and our goal is to label each segment with appropriate labels. We try to over-segment, so that we end up with more segments and avoid merging two sub-activities into one segment. Each of these segments occupies a small length of time and therefore, considering nodes per segment gives us a meaningful and concise representation for the graph \mathcal{G} . With such a representation, we can model meaningful transitions of a sub-activity following another, e.g., *pouring* followed by *moving*.

3.2 Overview of Properties Captured

Over the course of a video, a human may interact with several objects and perform several sub-activities over time. In our MRF model (in Section 3.3), these interactions are represented by the edges \mathcal{E} , and we try to capture the following properties:

Affordance - sub-activity relations. At any given time, the affordance of the object depends on the sub-activity it is involved in. For example, a cup has the affordance of ‘*pourable*’ in a *pouring* sub-action and has the affordance of ‘*drinkable*’ in a *drinking* sub-action. We compute relative geometric features between the object and the human’s skeletal joints to capture this.

Affordance - affordance relations. Objects have affordances even if they are not interacted directly with by the human, and their affordances depend on the affordances of other objects around them. E.g., in the case of *pouring* from a pitcher to a cup, the cup is not interacted by the human directly but has the affordance ‘*pour-to*’. We therefore use relative geometric features such as “on top of”, “nearby”, “in front of”, etc., to model the affordance - affordance relations.

Sub-activity change over time. Each activity consists of a sequence of sub-activities that change over the course of performing the activity. We model this by incorporating temporal edges in \mathcal{G} .

Affordance change over time. The object affordances depend on the activity being performed and hence change along with the sub-activity over time. We model the temporal change in affordances of each object using features such as change in appearance and location of the object over time.

3.3 Model

We model the spatio-temporal structure of an activity using a model isomorphic to a Markov Random Field with log-linear node and pairwise edge potentials (see Fig. 2 for an illustration). Let K_a denote the set of sub-activity labels, and K_o denote the set of object affordance labels. Given a temporally segmented 3D video $\mathbf{x} = (x_1, \dots, x_N)$ consisting of temporal segments x_s , we aim to predict a labeling $\mathbf{y} = (y_1, \dots, y_N)$ for each segment. For a segmented 3D video \mathbf{x} , the prediction $\hat{\mathbf{y}}$ is computed as the argmax of a discriminant function $f_{\mathbf{w}}(\mathbf{x}, \mathbf{y})$ that is parameterized by weights \mathbf{w} .

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} f_{\mathbf{w}}(\mathbf{x}, \mathbf{y}) \quad (1)$$

The discriminant function captures the dependencies between the sub-activity and object affordance labels as defined by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We now describe the structure of this graph. For object nodes denoted by \mathcal{V}_o and sub-activity nodes denoted by \mathcal{V}_a , let \mathcal{V}_o^s denote set of object nodes of segment s , and v_a^s denote the sub-activity node of segment s . For all segments s , there is an edge connecting all the nodes in \mathcal{V}_o^s to each other and to the sub-activity node v_a^s . These edges signify the relationships within the objects, and between the objects and the human pose within a segment and are referred to as ‘*object - object interactions*’ and ‘*sub-activity - object interactions*’ in the Fig. 2 respectively.

The sub-activity node of segment s is connected to the sub-activity nodes in segments $(s - 1)$ and $(s + 1)$. Similarly every object node of segment s is connected to the corresponding object nodes in segments $(s - 1)$ and $(s + 1)$. These edges model the *temporal interactions* between the human poses and the objects respectively and represented by dotted edges in the Fig. 2.

We categorize the edges into different types denoted by \mathcal{T} . The various types are *object - object* edges and *object - sub-activity* edges and *sub-activity - sub-activity* edges. Let \mathcal{E}_t denote the edges of type $t \in \mathcal{T}$ and T_t denote pairs of possible labels for the nodes connected by edges of type t . For example if the edge type t is object - sub-activity, then $(l, k) \in T_t, \forall l \in K_o, \forall k \in K_a$.

Let y_i^k be a binary variable representing the node i having label k , where $k \in K_o$ for object nodes and $k \in K_a$ for sub-activity nodes. All k binary variables together represent the label of a node in the above described graph \mathcal{G} . A segment label \mathbf{y}_s is composed of two types of node labels: a set of object affordance labels, $\{y_i^k : k \in K_o; \forall i \in \mathcal{V}_o^s\}$, and a sub-activity label $\{y_i^k : k \in K_a; i = v_a^s\}$. Given \mathcal{G} , we define the following discriminant function based on individual node features $\phi_a(i)$ and $\phi_o(i)$ and edge features $\phi_t(i, j)$ as further described below.

$$\begin{aligned} f_{\mathbf{w}}(\mathbf{y}, \mathbf{x}) = & \sum_{i \in \mathcal{V}_a} \sum_{k \in K_a} y_i^k [w_a^k \cdot \phi_a(i)] + \sum_{i \in \mathcal{V}_o} \sum_{k \in K_o} y_i^k [w_o^k \cdot \phi_o(i)] \\ & + \sum_{t \in \mathcal{T}} \sum_{(i, j) \in \mathcal{E}_t} \sum_{(l, k) \in T_t} y_i^l y_j^k [w_t^{lk} \cdot \phi_t(i, j)] \end{aligned} \quad (2)$$

For the node feature maps, $\phi_a(i)$ and $\phi_o(i)$, there is one weight vector for each of the sub-activity classes in K_a and each of the object affordance classes in K_o respectively. There are multiple types t of the edge feature maps $\phi_t(i, j)$, each corresponding to a different type of edge in the graph. For each type, there is one weight vector each for every pair of labels the edge can take.

Features. For a given object node i , the node feature map $\phi_o(i)$ is a vector of features representing the object’s location in the scene and how it changes within the temporal segment. These features include the (x, y, z) coordinates of the object’s centroid and the coordinates of the object’s bounding box at the middle frame of the temporal segment. We also run a SIFT feature based object tracker [25] to find the corresponding points between the adjacent frames and then compute the transformation matrix based on the matched image points. We add the transformation matrix corresponding to the object in the middle frame with respect to its previous frame to the features in order to capture the object’s motion information. In addition to the above features, we also compute the total displacement and the total distance moved by the object’s centroid in the set of frames belonging to the temporal segment. We then perform cumulative binning of the feature values into 10 bins. In our experiments, we have $\phi_o(i) \in \mathbb{R}^{180}$.

Similarly, for a given sub-activity node i , the node feature map $\phi_a(i)$ gives a vector of features computed using the human skeleton information obtained from running OpenNI’s skeleton tracker [26] on the RGBD video. We compute the features described above for each the upper-skeleton joint (neck, torso, left shoulder, left elbow, left palm, right shoulder, right elbow and right palm) locations relative to the subject’s head location, thus giving us $\phi_a(i) \in \mathbb{R}^{1030}$.

The edge feature maps $\phi_t(i, j)$ describe the relationship between node i and j . For capturing the *object-object* relations within a temporal segment, we compute relative geometric features such as the difference in (x, y, z) coordinates of the object centroids and the distance between them. These features are computed at the first, middle and last frames of the temporal segment along with *min* and *max* of their values across all frames in the temporal segment to capture the relative motion information. This gives us $\phi_1(i, j) \in \mathbb{R}^{200}$. Similarly for *object-sub-activity* relation features $\phi_2(i, j) \in \mathbb{R}^{400}$, we use the same features as for the *object-object* relation features, but we compute them between the upper-skeleton joints and the each object’s centroid. The temporal relational features capture the change across temporal segments and we use the vertical change in position and the distance between by corresponding object and the joint locations. This gives us $\phi_3(i, j) \in \mathbb{R}^{40}$ and $\phi_4(i, j) \in \mathbb{R}^{160}$ respectively.

Inference. Given the model parameters w , the inference problem is to find the best labeling $\hat{\mathbf{y}}$ for a new video \mathbf{x} , i.e., solving the argmax in Eq. (1) for the discriminant function in Eq. (2). This is a NP hard problem. However, its equivalent formulation as the following mixed-integer program has a linear relaxation which can be solved efficiently as a quadratic pseudo-Boolean optimization problem using a graph-cut method [27].

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} \max_{\mathbf{z}} \sum_{i \in \mathcal{V}_a} \sum_{k \in K_a} y_i^k [w_a^k \cdot \phi_a(i)] + \sum_{i \in \mathcal{V}_o} \sum_{k \in K_o} y_i^k [w_o^k \cdot \phi_o(i)] + \sum_{t \in \mathcal{T}} \sum_{(i,j) \in \mathcal{E}_t} \sum_{(l,k) \in \mathcal{T}_t} z_{ij}^{lk} [w_t^{lk} \cdot \phi_t(i,j)] \quad (3)$$

$$\forall i, j, l, k: z_{ij}^{lk} \leq y_i^l, z_{ij}^{lk} \leq y_j^k, y_i^l + y_j^k \leq z_{ij}^{lk} + 1, z_{ij}^{lk}, y_i^l \in \{0, 1\} \quad (4)$$

Note that the products $y_i^l y_j^k$ have been replaced by auxiliary variables z_{ij}^{lk} . Relaxing the variables z_{ij}^{lk} and y_i^l to the interval $[0, 1]$ results in a linear program that can be shown to always have half-integral solutions (i.e., y_i^l only take values $\{0, 0.5, 1\}$ at the solution) [28]. Since every node in our experiments has exactly one class label, we also consider the linear relaxation from above with the additional constraints $\forall i \in \mathcal{V}_a: \sum_{l \in K_a} y_i^l = 1$ and $\forall i \in \mathcal{V}_o: \sum_{l \in K_o} y_i^l = 1$. This problem can no longer be solved via graph cuts. We compute the exact mixed integer solution including these additional constraint using a general-purpose MIP solver¹ during inference. The MIP solver takes 10.7 seconds on an average for one video (a typical video has a graph with 17 sub-activity nodes and 592 object nodes, i.e., 6090 variables).

Learning. We take a large-margin approach to learning the parameter vector \mathbf{w} of Eq. (2) from labeled training examples $(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_M, \mathbf{y}_M)$ [29, 30]. Our method optimizes a regularized upper bound on the training error $R(h) = \frac{1}{M} \sum_{m=1}^M \Delta(\mathbf{y}_m, \hat{\mathbf{y}}_m)$, where $\hat{\mathbf{y}}_m$ is the optimal solution of Eq. (1) and $\Delta(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i \in \mathcal{V}_o} \sum_{k \in K_o} |y_i^k - \hat{y}_i^k| + \sum_{i \in \mathcal{V}_a} \sum_{k \in K_a} |y_i^k - \hat{y}_i^k|$. To simplify notation, note that Eq. (3) can be equivalently written as $\mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y})$ by appropriately stacking the w_a^k, w_o^k and w_t^{lk} into \mathbf{w} and the $y_i^k \phi_a(i), y_i^k \phi_o(i)$ and $z_{ij}^{lk} \phi_t(i, j)$ into $\Psi(\mathbf{x}, \mathbf{y})$, where each z_{ij}^{lk} is consistent with Eq. (4) given \mathbf{y} . Training can then be formulated as the following convex quadratic program [31]:

$$\begin{aligned} \min_{\mathbf{w}, \xi} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C\xi \\ \text{s.t.} \quad & \forall \bar{\mathbf{y}}_1, \dots, \bar{\mathbf{y}}_M \in \{0, 0.5, 1\}^{N \cdot K}: \frac{1}{M} \mathbf{w}^T \sum_{m=1}^M [\Psi(\mathbf{x}_m, \mathbf{y}_m) - \Psi(\mathbf{x}_m, \bar{\mathbf{y}}_m)] \geq \Delta(\mathbf{y}_m, \bar{\mathbf{y}}_m) - \xi \end{aligned} \quad (5)$$

While the number of constraints in this QP is exponential in M, N and K , it can nevertheless be solved efficiently using the cutting-plane algorithm [31]. The algorithm needs access to an efficient method for computing

$$\bar{\mathbf{y}}_m = \underset{\mathbf{y} \in \{0, 0.5, 1\}^{N \cdot K}}{\operatorname{argmax}} [\mathbf{w}^T \Psi(\mathbf{x}_m, \mathbf{y}) + \Delta(\mathbf{y}_m, \mathbf{y})]. \quad (6)$$

Due to the structure of $\Delta(\cdot, \cdot)$, this problem is identical to the relaxed prediction problem in Eqs. (3)-(4) and can be solved efficiently using graph cuts.

3.4 Multiple Segmentations

Segmenting an RGB-D video in time can be noisy, and multiple segmentations may be valid. Therefore, we will perform multiple segmentations by using different methods and criterion of segmentation (see Section 4 for details). Thus, we get a set \mathcal{H} of multiple segmentations, and let h_n be the n^{th} segmentation. A discriminant function $f_{\mathbf{w}^{h_n}}(\mathbf{x}^{h_n}, \mathbf{y}^{h_n})$ can now be defined for each h_n as in Eq. (2). We now define a score function $g_\theta(\mathbf{y}^{h_n}, \mathbf{y})$ which gives a score for assigning the labels of the segments from \mathbf{y}^{h_n} to \mathbf{y} ,

$$g_{\theta_n}(\mathbf{y}^{h_n}, \mathbf{y}) = \sum_{k \in K} \sum_{i \in \mathcal{V}} \theta_n^k \mathbf{y}_i^{h_n k} \mathbf{y}_i^k \quad (7)$$

where $K = K_s \cup K_a$. Here, θ_n^k can be interpreted as the confidence of labeling the segments of label k correctly in the n^{th} segmentation hypothesis. We want to find the labeling that maximizes the assignment score across all the segmentations. Therefore we can write inference in terms of a joint objective function as follows

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} \max_{\mathbf{y}^{h_n} \forall h_n \in \mathcal{H}} \sum_{h_n \in \mathcal{H}} [f_{\mathbf{w}^{h_n}}(\mathbf{x}^{h_n}, \mathbf{y}^{h_n}) + g_\theta(\mathbf{y}^{h_n}, \mathbf{y})] \quad (8)$$

¹<http://www.tfinley.net/software/pyglpk/readme.html>

Table 1: Results on Cornell 60 Activity Dataset [23], tested on “New Person” data for 12 activity classes.

	bathroom		bedroom		kitchen		living room		office		Average	
	prec	rec	prec	rec	prec	rec	prec	rec	prec	rec	prec	rec
Sung et. al. [23]	72.7	65.0	76.1	59.2	64.4	47.9	52.6	45.7	73.8	59.8	67.9	55.5
Our method	88.9	61.1	73.0	66.7	96.4	85.4	69.2	68.7	76.7	75.0	80.8	71.4

This formulation is equivalent to considering the labelings y^{h_n} over the segmentations as unobserved variables. It is possible to use the latent structural SVM [32] to solve this, but it becomes intractable if the size of the segmentation hypothesis space is large. Therefore we propose an approximate two-step learning procedure to address this. For a given set of segmentations \mathcal{H} , we first learn the parameters \mathbf{w}^{h_n} independently as described in Section 3.3. We then train the parameters θ on a separate held-out training dataset. This can now be formulated as a QP:

$$\min_{\theta} \quad \frac{1}{2} \theta^T \theta - \sum_{h_n \in \mathbf{H}} g_{\theta_n}(\mathbf{y}^{h_n}, \mathbf{y}) \text{ s.t. } \forall k \in K : \sum_{n=1}^{|\mathbf{H}|} \theta_n^k = 1$$

Using the fact that the objective function defined in Eq. (8) is convex, we design an iterative two-step procedure where we solve for $\mathbf{y}^{h_n}, \forall h_n \in \mathbf{H}$ in parallel and then solve for \mathbf{y} . This method is guaranteed to converge, and when the number of variables scales linearly with the number of segmentation hypothesis considered, the original problem in Eq. (8) will become considerably slow, but our method will still scale. More formally, we iterate between the following two problems:

$$\hat{\mathbf{y}}^{h_n} = \underset{\mathbf{y}^{h_n}}{\operatorname{argmax}} f_{\mathbf{w}^{h_n}}(\mathbf{x}^{h_n}, \mathbf{y}^{h_n}) + g_{\theta_n}(\mathbf{y}^{h_n}, \hat{\mathbf{y}}) \quad (9), \quad \hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} g_{\theta_n}(\hat{\mathbf{y}}^{h_n}, \mathbf{y}) \quad (10)$$

High-level Activity Classification. For classifying the high-level activity, we compute the histograms of sub-activity and affordance labels and use them as features. Since the occlusion of objects also plays a major role in some activities, we capture this by including in our feature vector, the fraction of objects that are occluded fully or partially in the temporal segments. We then train a multi-class SVM classifier on training data using these features.

4 Experiments

Data. We test our model on two 3D activity datasets: Cornell 60 [23] and Cornell 120 Activity Datasets. The Cornell 60 Activity Dataset [23] contains 60 3D videos of four different subjects performing 12 high-level activity classes. However, some of these activity classes contain only one sub-activity (e.g., *working on a computer*, *cooking (stirring)*, etc.) and do not contain object interactions (e.g., *talking on couch*, *relaxing on couch*).

We introduce the Cornell 120 Activity Dataset dataset, which contains activity sequences of ten different high-level activities performed by four different subjects, where each high-level activity was performed three times. We thus have 61,585 total 3D video frames in our dataset. The high-level activities are: {*making cereal*, *taking medicine*, *stacking objects*, *unstacking objects*, *microwaving food*, *picking objects*, *cleaning objects*, *taking food*, *arranging objects*, *having a meal*}. The subjects were only given a high-level description of the task,² and were asked to perform the activities multiple times with *different* objects. For example, the stacking and unstacking activities were performed with pizza boxes, plates and bowls. They performed the activities through a long sequence of sub-activities, which varied from subject to subject significantly in terms of length of the sub-activities, order of the sub-activities as well as in the way they executed the task. The camera was mounted so that the subject was in view (although the subject may not be facing the camera), but often there were significant occlusions of the body parts. See Fig. 1 for some examples.

We labeled our dataset with the sub-activity and the object affordance labels. We refer to the set of contiguous frames spanning a sub-activity as a temporal segment. Specifically, our sub-activity labels are: {*reaching*, *moving*, *pouring*, *eating*, *drinking*, *opening*, *placing*, *closing*, *scrubbing*, *null*} and our affordance labels are: {*reachable*, *movable*, *pourable*, *pourto*, *containable*, *drinkable*, *openable*, *placeable*, *closable*, *scrubbable*, *scrubber*, *stationary*}. Table 1 in supplementary material shows the details on which sub-activities are present in each high-level activity.

Preprocessing. Given the raw data containing the color and depth values for every pixel in the video, we first tracked the human skeleton using Openni’s skeleton tracker [26] for obtaining the locations of the various joints of the human skeleton. However these values are not very accurate, as the Openni’s skeleton tracker is only designed to track human skeletons in clutter-free environments and without any occlusion of the body parts. In real-world human activity videos, some body parts are often occluded and the interaction with the objects hinders accurate skeleton tracking. We show

²For example, the instructions for *making cereal* were: 1) Place bowl on table, 2) Pour cereal, 3) Pour milk. For *microwaving food*, they were: 1) Open microwave door, 2) Place food inside, 3) Close microwave door.

Table 2: Results on Cornell 120 Activity Dataset, showing average micro precision/recall, and average macro precision and recall for affordance, sub-activities and high-level activities. Standard error is also reported.

method	Object Affordance			Sub-activity			High-level Activity		
	micro		macro	micro		macro	micro		macro
	P/R	Prec.	Recall	P/R	Prec.	Recall	P/R	Prec.	Recall
<i>max class</i>	65.7 \pm 1.0	65.7 \pm 1.0	8.3 \pm 0.0	29.2 \pm 0.2	29.2 \pm 0.2	10.0 \pm 0.0	10.0 \pm 0.0	10.0 \pm 0.0	10.0 \pm 0.0
<i>image only</i>	74.2 \pm 0.7	15.9 \pm 2.7	16.0 \pm 2.5	56.2 \pm 0.4	39.6 \pm 0.5	41.0 \pm 0.6	34.7 \pm 2.9	24.2 \pm 1.5	35.8 \pm 2.2
<i>SVM multiclass</i>	75.6 \pm 1.8	40.6 \pm 2.4	37.9 \pm 2.0	58.0 \pm 1.2	47.0 \pm 0.6	41.6 \pm 2.6	30.6 \pm 3.5	27.4 \pm 3.6	31.2 \pm 3.7
<i>MEMM[23]</i>	-	-	-	-	-	-	26.4 \pm 2.0	23.7 \pm 1.0	23.7 \pm 1.0
<i>object only</i>	86.9 \pm 1.0	72.7 \pm 3.8	63.1 \pm 4.3	-	-	-	59.7 \pm 1.8	56.3 \pm 2.2	58.3 \pm 1.9
<i>sub-activity only</i>	-	-	-	71.9 \pm 0.8	60.9 \pm 2.2	51.9 \pm 0.9	27.4 \pm 5.2	31.8 \pm 6.3	27.7 \pm 5.3
<i>no temporal interactions</i>	87.0 \pm 0.8	79.8 \pm 3.6	66.1 \pm 1.5	76.0 \pm 0.6	74.5 \pm 3.5	66.7 \pm 1.4	81.4 \pm 1.3	83.2 \pm 1.2	80.8 \pm 1.4
<i>no object interactions</i>	88.4 \pm 0.9	75.5 \pm 3.7	63.3 \pm 3.4	85.3 \pm 1.0	79.6 \pm 2.4	74.6 \pm 2.8	80.6 \pm 2.6	81.9 \pm 2.2	80.0 \pm 2.6
<i>full model: groundtruth seg</i>	91.8 \pm 0.4	90.4 \pm 2.5	74.2 \pm 3.1	86.0 \pm 0.9	84.2 \pm 1.3	76.9 \pm 2.6	84.7 \pm 2.4	85.3 \pm 2.0	84.2 \pm 2.5

Full model. End-to-end results, *without* assuming any ground-truth temporal segmentation is given.

<i>full, 1 segment. (best)</i>	83.1 \pm 1.1	70.1 \pm 2.3	63.9 \pm 4.4	66.6 \pm 0.7	62.0 \pm 2.2	60.8 \pm 4.5	77.5 \pm 4.1	80.1 \pm 3.9	76.7 \pm 4.2
<i>full, 1 segment. (averaged)</i>	81.3 \pm 0.4	67.8 \pm 1.1	60.0 \pm 0.8	64.3 \pm 0.7	63.8 \pm 1.1	59.1 \pm 0.5	79.0 \pm 0.9	81.1 \pm 0.8	78.3 \pm 0.9
<i>full, multi-seg learning</i>	83.9 \pm 1.5	75.9 \pm 4.6	64.2 \pm 4.0	68.2 \pm 0.3	71.1 \pm 1.9	62.2 \pm 4.1	80.6 \pm 1.1	81.8 \pm 2.2	80.0 \pm 1.2

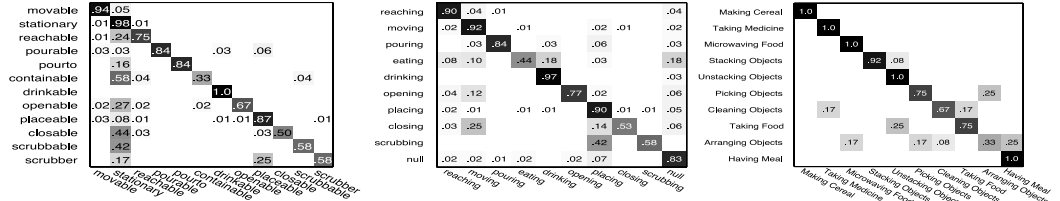


Figure 3: Confusion matrix for affordance labeling (left), sub-activity labeling (middle) and high-level activity labeling (right) of the test RGB-D videos.

that even with such noisy data, our method gets high accuracies by modeling the mutual context between the affordances and sub-activities.

We then use SIFT feature matching [25], while enforcing depth consistency across the time frames for obtaining reliable object tracks in the 3D video. In our current implementation, we need to provide the bounding box of the objects involved in the activity in the first frame of the video. Note that these are just bounding boxes and not object labels. In future, such bounding boxes of objects of interest can also be obtained in a preprocessing step by either running a set of object detectors [33] or some other methods that use 3D information [34, 35]. The outputs from the skeleton and object tracking along with the RGBD videos are then used to generate the features.

Labeling results on the Cornell 60 Activity Dataset. Table 1 shows the precision and recall of the high-level activities on the Cornell Activity Dataset [23]. Following Sung et. al.’s [23] experiments, we considered the same five groups of activities based on their location, and learnt a separate model for each location. To make it a fair comparison, we do not assume perfect segmentation of sub-activities and do not use any object information. Therefore, we train our model with only sub-activity nodes and consider segments of uniform size (20 frames per segments). We consider only a subset of our features described in Section 3.3 that are possible to compute from the tracked human skeleton and RGBD data provided in this dataset. Table 1 shows that our model significantly outperforms Sung et. al.’s MEMM model even when using only the sub-activity nodes and a simple segmentation algorithm. More detailed results are provided in the supplementary material.

Labeling results on Cornell 120 Activity Dataset. Table 2 shows the performance of various models on object affordance, sub-activity and high-level activity labeling. These results are obtained using 4-fold cross-validation and averaging performance across the folds. Each fold constitutes the activities performed by one subject, therefore the model is trained on activities of three subjects and *tested on a new subject*. We report both the micro and macro averaged precision and recall over various classes along with standard error. Since our algorithm can only predict one label for each segment, micro precision and recall are same as the percentage of correctly classified segments. Macro precision and recall are the averages of precision and recall respectively for all classes.

Assuming ground-truth temporal segmentation is given, the results for our *full model* are shown in Table 2 on line 9, its variations on lines 5-8 and the baselines on lines 1-3. The results in lines 10-12 correspond to the case when temporal segmentation is not assumed. In comparison to a basic SVM multiclass model [31] (referred to as *SVM multiclass* when using all features and *image only* when using only image features), which is equivalent to only considering the nodes in our MRF without any edges, our model performs significantly better. We also compare with the high-level activity classification results obtained from the method presented in [23]. We ran their code on our dataset and obtain accuracy of 26.4%, whereas our method gives an accuracy of 84.7% when

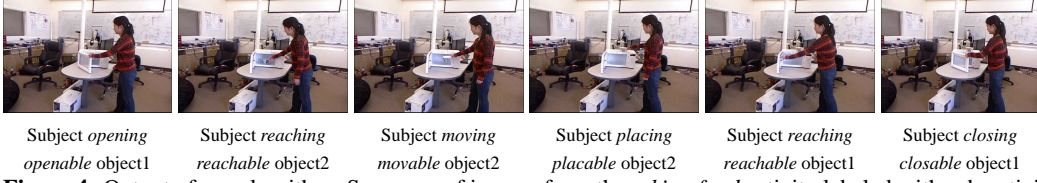


Figure 4: Output of our algorithm: Sequence of images from the *taking food* activity labeled with sub-activity and object affordance labels.

ground truth segmentation is available and 80.6% otherwise. Figure 4 shows a sequence of images from *taking food* activity along with the inferred labels. Figure 3 shows the confusion matrix for labeling affordances, sub-activities and high-level activities with our proposed method. We can see that there is a strong diagonal with a few errors such as *scrubbing* misclassified as *placing*, and *picking objects* misclassified as *arranging objects*.

We analyze our model to gain insight into which interactions provide useful information by comparing our full model to variants of our model.

How important is object context for activity detection? We show the importance of object context for sub-activity labeling by learning a variant of our model without the object nodes (referred to as *sub-activity only*). With object context, the micro precision increased by 14.1% and both macro precision and recall increased by around 23.3% over *sub-activity only*. Considering object information (affordance labels and occlusions) also improved the high-level activity accuracy by 3-fold.

How important is activity context for affordance detection? We also show the importance of context from sub-activity for affordance detection by learning our model without the sub-activity nodes (referred to as *object only*). With sub-activity context, the micro precision increased by 4.9% and the macro precision and recall increased by 17.7% and 11.1% respectively for affordance labeling over *object only*. The relative gain is less compared to that obtained in sub-activity detection as the *object only* model still has object-object context which helps in affordance detection.

How important is object - object context for affordance detection? In order to study the effect of the object-object interactions for affordance detection, we learnt our model without the object-object edge potentials (referred to as *no object interactions*). We see a considerable improvement in affordance detection when the object interactions are modeled, the macro recall increased by 14.9% and the macro precision by about 10.9%. This shows that sometimes just the context from the human activity alone is not sufficient to determine the affordance of an object.

How important is temporal context? We also learn our model without the temporal edges (referred to as *no temporal interactions*). Modeling temporal interactions increased the micro precision by 4.8% and 10.0% for affordances and sub-activities respectively and increased the micro precision for high-level activity by 3.3%.

End-to-end Results. Given the RGB-D video, we obtain the final labeling using our method described in Section 3.4. To generate the segmentation hypothesis set \mathcal{H} we consider three different segmentation algorithms, and generate multiple segmentations by changing their parameters. We consider: 1) uniform segmentation parameterized by segment size, 2) graph based segmentation [36] with edge weights defined using the displacement of skeletal joints, and 3) graph based segmentation with edge weights defined using rate of change of skeletal joint locations. The lines 10-12 of Table 2 show the results of the best performing segmentation, average performance all the segmentations considered, and our proposed method for combining the segmentations. We see that our method improves the performance over considering a single best performing segmentation: macro precision increased by 5.8% and 9.1% for affordance and sub-activity labeling respectively.

5 Conclusion

In this paper, we considered the task of jointly labeling human activities and object affordances from RGB-D videos. The activities we consider happen over a long time period, and comprise several sub-activities performed in a sequence. We formulated this problem as a Markov Random Field, and learned the parameters of the model using a structural SVM formulation. Our model also incorporates the temporal segmentation problem by computing several segmentations and considering labeling over these segmentations as latent variables. In extensive experiments over a challenging dataset, we show that our method achieves an end-to-end accuracy precision of 81.8% and recall of 80.0% for labeling the activities performed by a different subject than the ones in the training set. We also showed that it is important to model the different properties (object affordances, object-object interaction, temporal interactions, etc.) in order to achieve good performance.

References

- [1] A. Gupta, A. Kembhavi, and L.S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE T-PAMI*, 31(10):1775–1789, 2009.
- [2] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010.
- [3] J.J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.
- [4] P. Kormushev, S. Calinon, and D. G. Caldwell. Robot motor skill coordination with EM-based reinforcement learning. In *IROS*, 2010.
- [5] T. Finley and T. Joachims. Training structural svms when exact inference is intractable. In *ICML*, 2008.
- [6] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *CVPR*, 2010.
- [7] B. Yao, X. Jiang, A. Khosla, A.L. Lin, L.J. Guibas, and L. Fei-Fei. Action recognition by learning bases of action attributes and parts. In *ICCV*, 2011.
- [8] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos “in the wild”. In *CVPR*, 2009.
- [9] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.
- [10] S. Sadanand and J.J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012.
- [11] Q. Shi, L. Wang, L. Cheng, and A. Smola. Human action segmentation and recognition using discriminative semi-markov models. *IJCV*, 2011.
- [12] M. Hoai, Z. Lan, and F. De la Torre. Joint segmentation and classification of human actions in video. In *CVPR*, 2011.
- [13] P. Matikainen, R. Sukthankar, and M. Hebert. Model recommendation for action recognition. In *CVPR*, 2012.
- [14] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012.
- [15] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012.
- [16] H. Pirsivash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, 2012.
- [17] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Comp Surveys (CSUR)*, 2011.
- [18] E.E. Aksoy, A. Abramov, F. Worgotter, and B. Dellen. Categorizing object-action relations from semantic scene graphs. In *ICRA*, 2010.
- [19] Y. Jiang, Z. Li, and S. Chang. Modeling scene and object contexts for human action retrieval with few examples. *IEEE Trans Circuits & Sys Video Tech*, 2011.
- [20] H. Kjellström, J. Romero, and D. Kragic. Visual object-action recognition: Inferring object affordances from human demonstration. *CVIU*, 115(1):81–90, 2011.
- [21] H. Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. In *NIPS*, 2011.
- [22] K. Lai, L. Bo, X. Ren, and D. Fox. Sparse Distance Learning for Object Recognition Combining RGB and Depth Information. In *ICRA*, 2011.
- [23] J. Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured human activity detection from rgbd images. In *ICRA*, 2012.
- [24] J. Gall, A. Fossati, and L. van Gool. Functional categorization of objects using real-time markerless motion capture. In *CVPR*, 2011.
- [25] O. Pele and M. Werman. A linear time histogram metric for improved sift matching. In *ECCV*, 2008.
- [26] <http://openpi.org>.
- [27] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. In *CVPR*, 2007.
- [28] P.L. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation and persistency in quadratic 0–1 optimization. *Mathematical Prog.*, 28(2):121–155, 1984.
- [29] B. Taskar, V. Chatalbashev, and D. Koller. Learning associative markov networks. In *ICML*, 2004.
- [30] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.
- [31] T. Joachims, T. Finley, and C. Yu. Cutting-plane training of structural SVMs. *Mach. Learn.*, 77(1), 2009.
- [32] C. Yu and T. Joachims. Learning structural svms with latent variables. In *ICML*, 2009.
- [33] L.J. Li, H. Su, E.P. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *NIPS*, 2010.
- [34] M. Martinez, A. Collet, and S.S. Srinivasa. Moped: A scalable and low latency object recognition and pose estimation system. In *ICRA*, 2010.
- [35] R.B. Rusu, N. Blodow, Z. Marton, A. Soos, and M. Beetz. Towards 3d object maps for autonomous household robots. In *IROS*, 2007.
- [36] P. F. Felzenszwalb and D.P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2), 2004.